

MEGA: merger graphs of structure formation

Article (Published Version)

Roper, William J, Thomas, Peter A and Srisawat, Chaichalit (2020) MEGA: merger graphs of structure formation. *Monthly Notices of the Royal Astronomical Society*, 494 (3). pp. 4509-4524. ISSN 0035-8711

This version is available from Sussex Research Online: <http://sro.sussex.ac.uk/id/eprint/90689/>

This document is made available in accordance with publisher policies and may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the URL above for details on accessing the published version.

Copyright and reuse:

Sussex Research Online is a digital repository of the research output of the University.

Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable, the material made available in SRO has been checked for eligibility before being made available.

Copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.



MEGA: Merger graphs of structure formation

William J. Roper¹, ¹★ Peter A. Thomas¹ and Chaichalit Srisawat²

¹*Astronomy Centre, University of Sussex, Falmer, Brighton BN1 9QH, UK*

²*Center for Astrophysics and Cosmology, Science Institute, University of Iceland, Dunhagi 5, 107 Reykjavik, Iceland*

Accepted 2020 April 2. Received 2020 April 1; in original form 2020 February 28

ABSTRACT

When following the growth of structure in the Universe, we propose replacing merger trees with merger graphs, in which haloes can both merge and split into separate pieces. We show that this leads to smoother mass growth and eliminates catastrophic failures in which massive haloes have no progenitors or descendants. For those who prefer to stick with merger trees, we find that trees derived from our merger graphs have similar mass growth properties to previous methods, but again without catastrophic failures. For future galaxy formation modelling, two different density thresholds can be used to distinguish host haloes (extended galactic haloes, groups, and clusters) from higher density subhaloes: sites of galaxy formation.

Key words: methods: numerical – galaxies: evolution – galaxies: haloes.

1 INTRODUCTION

Structure formation in the Universe is hierarchical in nature, with small haloes forming first and merging into ever-larger ones in a process that is usually visualized in the form of a merger tree, first popularized by Lacey & Cole (1993).

The Sussing Merger Trees project (Srisawat et al. 2013, hereafter SMT13) applied 10 different merger tree algorithms on the same dark matter halo catalogue and compared the results. These all showed a variety of issues characterized by extreme fluctuations in mass along the merger history of a halo and the occasional abrupt disappearance (i.e. lack of progenitors) of even quite massive haloes looking back in time. Most of these problems are caused by either a difficulty in locating the centre about which to grow a halo, leading to occasional ‘flip-flops’ in which the halo’s centre jumps from one substructure to another or the disappearance of a satellite subhalo as it passes through another, larger halo, and then its reappearance at a later stage.

A subsequent paper by Avila et al. (2014) investigated the extent to which the results of SMT13 were affected by the use of a particular halo finder. They used a wider range of halo-finding techniques and found that although the detailed statistics of halo growth were altered, the basic problems mentioned above remained.

This paper investigates the extent to which these problems can be overcome by the use of merger graphs rather than merger trees, in which haloes are allowed both to split apart and merge together: this fundamental difference between merger trees and the corresponding merger graphs is illustrated in Fig. 1. We will show that they eliminate catastrophic failures (appearance or disappearance of massive haloes) and provide smoother mass growth. Moreover, although that is not our proposed use, we find that such graphs

can be split after construction back into trees, whose properties are more well behaved than any investigated in SMT13.

One of the main uses of merger trees is to provide a skeleton of the evolution of dark matter haloes, within which to build models of galaxy formation. These models are compromised in existing trees by the flip-flop behaviour mentioned above. We suggest that the best way to circumvent that is not to divide substructures into, as existing methods do, a single main halo that is associated with the whole dark matter halo, and other subhaloes that are deemed to be satellites. Instead, we propose that all substructures are treated equally, but with the possibility of deeming one of those substructures to host a central galaxy if it is at, or close to, the dynamical centre of the enclosing halo.

We identify haloes with a friends-of-friends (FOF) linking algorithm. Unlike most existing algorithms, we choose not to define a centre about which to grow a spherical overdensity (SO). That not only introduces the problem of misidentification of main haloes, as mentioned above, but also leads to the possibility of overlapping haloes. Moreover, we identify substructures using FOF with a smaller linking length, which necessarily leads to the correct nesting of high-density substructures within lower density haloes. In the context of galaxy formation modelling, the low-density haloes would be container haloes (clusters, groups, or even extended dark matter haloes around individual galaxies) with the higher density substructures being the localized dark matter haloes within which galaxies form.

The paper is constructed as follows. In the rest of this introduction, we explain some of the terminology related to merger graphs and review previous work. In Section 2, we describe our MEGA (Merger Graph Algorithm) method for identifying haloes, constructing merger graphs, splitting those graphs back into trees, and nesting of subhaloes. Section 3 first contrasts our merger tree results with those from SMT13 and then shows the equivalent plots for merger graphs, before looking at the distribution and dynamics

* E-mail: w.roper@sussex.ac.uk

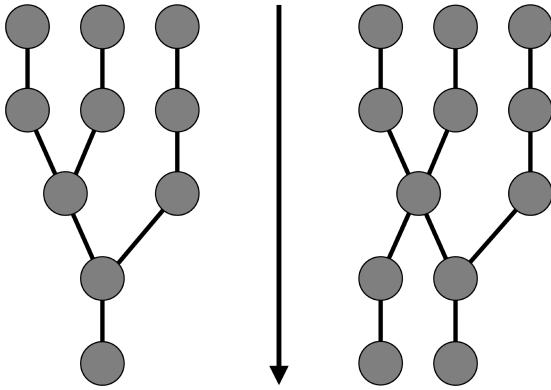


Figure 1. A schematic showing the differences between a merger tree (left) and a merger graph (right). The arrow represents the direction of time with the tree and graph rooted at the present day. The fundamental difference between a merger tree and a merger graph is that a merger graph permits haloes splitting apart, whereas a tree has a single descendent. This leads to potential extra branches in the graph, branching off towards the present day, as shown here.

of subhaloes. Finally, in Section 4, we present our conclusions and discuss their implications.

1.1 Terminology

Where possible, we stick to the terminology outlined in SMT13 and expanded upon in Thomas et al. (2015, hereafter TOT15). We provide a brief glossary of some key terms here. First,

(i) A halo is a dark matter condensation as returned by a halo finder.

There are a variety of different halo-finding algorithms in use: we describe the ones of relevance to this paper in Section 1.2 below.

(i) A temporal merger graph is a set of ordered halo pairs, (H_A, H_B) , where H_A is older than H_B , which represents the growth of structure over cosmic time.

(ii) Recursively, H_A itself and progenitors of H_A are progenitors of H_B . Where it is necessary to distinguish H_A from earlier progenitors, we will use the term direct progenitor.

(iii) Recursively, H_B itself and descendants of H_B are descendants of H_A . Where it is necessary to distinguish H_B from later descendants, we will use the term direct descendant.

(iv) Optionally, one of the direct progenitors may be labelled the main progenitor – in this paper, we will take that to be the most massive.

(v) The longest continuous sequence of main progenitors extending back in time from a given halo is known as the main branch.

(vi) A halo that has no descendants is known as an end halo.

(vii) A temporal merger tree is a temporal merger graph in which there is precisely one direct descendant for every halo, except for a single end halo.

The word temporal has been inserted into the definitions above because it is also possible to nest structure at any given epoch into a hierarchy of haloes and subhaloes of different overdensity. In this paper, we use only two density levels and so we use this restricted terminology:

(i) The spatial nesting of haloes is described by a set of ordered halo pairs, (H_A, H_B) , where H_A is nested within H_B .

(ii) H_A is a subhalo of H_B .

(iii) H_B is the host halo of H_A .

(iv) The subhalo closest to the centre of the host halo we will call the central subhalo – note that this differs from TOT15, who use the term main halo.

In addition to the above, we use a couple of terms to describe intermediate steps in the graph construction:

(i) A candidate halo has been selected for further investigation but is not yet confirmed as a true halo.

(ii) When producing trees from our graphs, for comparison with previous work, we often have to divide haloes into two or more pieces: we call each of these a split-halo.

1.2 Previous work

The first stage in developing robust and complete merger graphs or trees is to identify a halo catalogue. A comparison of existing techniques was described in the ‘Haloes Gone MAD’ project (Knebe et al. 2011). Those are primarily based on the FOF algorithm, either in real space or in phase space, and/or an SO step (Press & Schechter 1974) to grow haloes around density peaks. We choose the former method so as to ensure non-overlapping haloes.

As mentioned above, SMT13 and Avila et al. (2014) are two studies that look in detail at different merger tree algorithms and compare their properties. To the best of our knowledge, there have been no previous published papers on merger graphs of structure formation, except the above-mentioned TOT15 arXiv article that proposes a terminology for merger graphs as part of a longer article on merger tree data format. We note, however, that Han et al. (2012) produce a subhalo-level merger tree and as such implicitly produce a graph at host level when tracking fly-by and ejected subhaloes.

It is not our intention to provide a detailed description of either halo finding or tree-making algorithms in this paper as that is done in great detail in the above-mentioned papers. However, we do make comparison in this paper to much of that previous work, particularly SMT13 whose data we had access to, so we present a brief description of that here.

1.2.1 The simulation

In this work, we utilize the simulation produced during SMT13 to directly compare to their results. This cosmological dark matter-only simulation was performed using the GADGET-3 N -body code (Springel 2005) with initial conditions drawn from the WMAP-7 cosmology (Komatsu et al. 2011). It has 270^3 particles with a dark matter particle mass resolution of $9.31 \times 10^8 h^{-1} M_\odot$. The simulation was performed in a $(62.5 h^{-1} \text{ Mpc})^3$ box, with 62 snapshots covering a redshift range from $z = 50$ to $z = 0$. We had full access to the output of this simulation in addition to the data produced by the participating merger tree construction algorithms.

1.2.2 Halo finders

The Amiga Halo Finder (AHF) (Gill, Knebe & Gibson 2004; Knollmann & Knebe 2009) employs recursively adaptive grids to locate local overdensities in the density field. The identified density peaks are then treated as centres of prospective haloes. After identifying haloes, unbinding is done based on the SO model and halo properties are then calculated based on the particles asserted to be gravitationally bound to their respective density peak. This is

the halo finder employed by SMT13 and thus the halo finder we compare directly to.

ROCKSTAR (Behroozi, Wechsler & Wu 2013a) is a halo finder that builds a hierarchy of particles from FOF groups in phase space by progressively and adaptively reducing the linking length. Unbinding is then performed using the full particle potentials and subsequently halo centres are defined by averaging particle positions in a small region close to the phase-space density peak. Although merger tree data were not available for direct comparison, this halo finder is the closest in essence to the approach presented in this paper.

SUBFIND (Springel et al. 2001) identifies disjoint, gravitationally bound, locally, overdense regions within an input set of particles, traditionally provided by an FOF group finder. The algorithm then searches for saddle points in the iso-density contours within the global field of the halo to identify substructures. Then, the self-boundness condition is asserted so that the particles with positive total energy are iteratively dismissed until only bound particles remain.

1.2.3 Merger tree construction algorithms

Aside from JMERGE (developed as part of SMT13), which estimated the trajectory of objects, all participants in SMT13: MERGERTREE (part of AHF), CONSISTENT TREES (Behroozi et al. 2013b), D-TREES (Jiang et al. 2014), HBT (Han et al. 2012), LHALOTREE (Springel et al. 2005), SUBLINK (Rodriguez-Gomez et al. 2015), TREEMAKER (Tweed et al. 2009), VELOCIRAPTOR (Elahi et al. 2019a, b), and YSAMTM (Jung, Lee & Yi 2014), used particle IDs to link objects between snapshots. To make merger trees, all algorithms choose at most one object as the descendant of an older object and to do this, they use a variety of the available halo information. Some only count the fraction of shared particle IDs while others might use extra information such as position, velocity, binding energy, or any other physical properties of the particles or haloes. A summary of these approaches is presented in detail in SMT13.

Throughout the merger tree analysis presented here, we make comparisons to all the available SMT13 algorithms; however, data for comparison to HBT were unavailable for this paper, as such this algorithm is omitted here but presented in full in SMT13: we note that the omission of this algorithm does not effect the result and conclusions of this paper. For clarity, in subsequent analysis we focus on comparison to D-Trees due to its main branch length performance, being the next best to MEGA. For that reason, we present greater detail on D-Trees.

The D-Trees algorithm is designed to work with halo catalogues extracted by the SUBFIND halo finder and like AHF, this can lead to missing progenitors and descendants. To combat this, D-Trees allows for descendants to be identified multiple snapshots later, with a free parameter N_{step} defining how many snapshots are used for the descendant search; for SMT13, $N_{\text{step}} = 5$ was used.

These descendants are identified by finding a most bound ‘core’ of particles from the particles in all progenitors in snapshot A. A number of linking particles are then defined from the group of progenitors, a descendant is then any halo in snapshot B (where snapshot B is within snapshots A+1 to A+ N_{step}), which shares at least one of these linking particles. The descendant at snapshot B is then the candidate descendant with the maximum fraction of linking particles. This can yield up to N_{step} descendants. To identify the descendant used in the merger tree, the main progenitor of each

is found: if this main progenitor is the group at snapshot A for more than one of the descendants, then the earliest descendent is taken; if it is the main progenitor for only one descendent, then this descendent is taken; or if no such descendent exists, then the earliest descendent is taken regardless.

We note that some tree construction algorithms were designed to work with bespoke halo finders and, as such, the SMT13 comparison may not show them in their best light. This was investigated in Avila et al. (2014). Unfortunately, we do not have access to data from that paper and so cannot add those results to our plots for which the reader will have to refer back to the original paper. We note, however, that the improvement is modest compared to the results of our own MEGA algorithm that we describe below.

2 METHOD

In Section 2.1, we describe our MEGA method for identifying haloes: because of our desire to have distinct, non-overlapping haloes, that will be based solely on the FOF (Davis et al. 1985) algorithm together with a binding energy check. We then, in Section 2.2, describe the manner of linking those haloes into merger graphs. To facilitate comparison with previous work, we describe in Section 2.3 a method of splitting those graphs up into trees. Finally, in Section 2.4, we introduce the concept of spatial nesting of haloes of differing overdensity (i.e. FOF linking length).

Throughout this paper, we will use the same input simulations as in SMT13 and contrast our results to those from the various algorithms used in that paper.

2.1 Identification of haloes

We will first describe the identification of haloes in configuration space and then extend that to phase space, which we will show is better able to separate ‘fly-by’ haloes that have temporarily merged and eliminate spurious haloes. A flow chart summarizing the halo-finding algorithm is shown in Fig. B1 in Appendix B.

2.1.1 Spatially defined (configuration space) haloes

We generate candidate haloes by linking together all particles with separations less than

$$\ell_s = \alpha_s \bar{d}, \quad (1)$$

where \bar{d} is the mean inter-particle separation and $\alpha_s = 0.2$; this corresponds roughly to an overdensity with respect to the mean of 300–550 (More et al. 2011). Candidate haloes are required to contain a particle number $N_p \geq 10$. This minimum particle number is rather small and means that many of the smallest candidate haloes will be spurious; however, we want to maximize the chances that a larger halo, once it has formed, will survive until the present day (albeit that it may merge with a more massive halo).

For a candidate halo to make it into the final halo catalogue, we require one of the following to be true, whose motivation is explained below: either

- (i) The number of particles, $N_p \geq 20$ and
- (ii) The total energy of the halo, $E \leq 0$;

or

- (i) The halo has 10 or more particles in common with a halo in the previous snapshot.

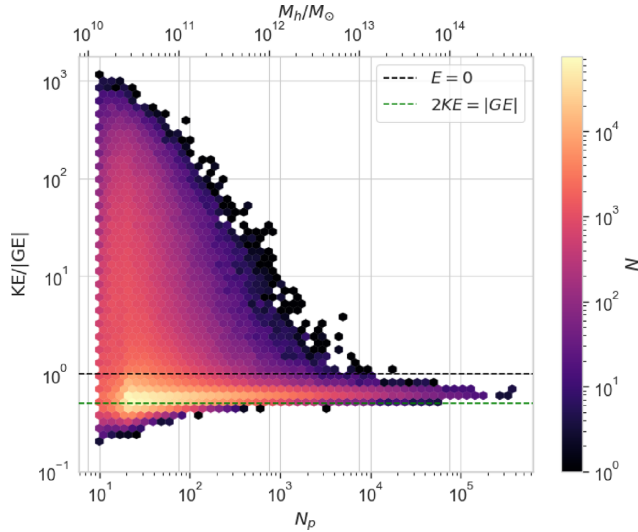


Figure 2. The energy distribution of spatially defined haloes with the ‘realness’ criterion applied as a function of their mass. The colour of the markers signifies the density of points in that region of the distribution. Where $N_p < 20$, haloes are included only if they track the evolution of a halo which at a previous time had $N_p \geq 20$ (i.e. was deemed real).

This ensures that each halo is at least 20 particles in size when it first appears, although it may fluctuate below that in its subsequent evolution.

The energy of a halo, E , is defined as

$$E = KE + GE, \quad (2)$$

where

$$KE = \frac{1}{2} M \sigma_v^2 \equiv \sum_{i=1}^{N_p} \frac{1}{2} m_i |\mathbf{v}_i - \langle \mathbf{v} \rangle|^2 \quad (3)$$

and

$$GE = - \sum_{i=1}^{N_p} \sum_{j=1}^{i-1} \frac{G m_i m_j}{(r_{ij}^2 + s^2)^{1/2}}. \quad (4)$$

Here, the sums run over the particles in a halo, $i = 1 \dots N_p$; m_i is the mass of particle i and M the total mass of the halo; \mathbf{v}_i is the particle velocity; $\langle \mathbf{v} \rangle$ the mean velocity of the halo; σ_v its (3-D) velocity dispersion; G is the gravitational constant; r_{ij} is the separation of particles i and j ; and s is the softening of the simulation.

Fig. 2 shows the value of E for each of our spatially defined haloes. The lower dashed line shows the relation expected for an isolated, virialized halo (whose size is much larger than the softening). The modal relation for our haloes lies slightly above this line principally because they are not isolated from their surroundings but are exchanging particles across the nominal outer density contour of the halo.

It is clear that a substantial fraction of haloes have positive energy. At low particle number, N_p , most of these are spurious associations of particles that have broken away from a larger halo and which are either subsequently reincorporated or rapidly disperse. It would be possible to weed these out based upon their future behaviour, and indeed we have a method to do this that we included in an earlier draft of this paper; however, we discovered that the phase-space linking described below does a much better job and so we decided to omit that more complicated spatial algorithm.

There are also haloes with very high particle number, $N_p > 1000$, even one with $N_p > 10000$, that have positive energy: these are merging systems whose orbital energy dominates over the binding energy of the merging systems.

Note that we do not consider whether individual particles are bound to the halo, which is at the heart of many other methods. That means that the outer parts of our FOF haloes may well contain particles that are not bound (and, conversely, there may be particles that lie outside the FOF halo that are gravitationally bound to it). There are two reasons for this: first, unbinding is an ill-defined and costly procedure, and secondly, it can lead to disruption, and hence disappearance from the graphs, of merging haloes. We find that the main positive effects of particle unbinding, namely identifying spurious haloes, or removing ‘fluff’ from the outskirts of haloes, can be accomplished by moving to phase-space linking, as described in the next subsection.

2.1.2 Phase space

To reduce the number of massive merging haloes with $E > 0$, which are in fact genuine interacting systems, while also reducing the number of temporary particle associations, we move to a phase-space halo definition. We do this by initially applying the spatial metric

$$\frac{r_{ij}^2}{\ell_s^2} \leq 1, \quad (5)$$

to produce candidate haloes, which are then individually passed through the phase-space metric

$$\frac{r_{ij}^2}{\ell_s^2} + \frac{v_{ij}^2}{\ell_v^2} \leq 2, \quad (6)$$

where r_{ij} is the particle separation in real space, v_{ij} is the particle separation in velocity space, ℓ_s is the previous spatial linking length (equation 1), and ℓ_v is the velocity space linking length.

We define an adaptive velocity space linking length derived from the virial theorem for an isolated halo of mass M , with particle mass m_p , number of particles N_p , and overdensity Δ relative to the mean density $\bar{\rho}$. We start with the virial theorem for an isolated isothermal sphere¹

$$2 \times \frac{1}{2} M \sigma_v^2 + \frac{G M^2}{2r} = 0, \quad (7)$$

where r is the radius of a sphere of density $\Delta \bar{\rho}$ and mass M ,

$$r = \left(\frac{3M}{4\pi \Delta \bar{\rho}} \right)^{1/3}. \quad (8)$$

Hence,

$$\sigma_v^2 = \frac{G}{2} \left(\frac{4\pi \Delta \bar{\rho} M^2}{3} \right)^{1/3} = \left(\frac{\Delta \bar{\rho} \pi G^3 (m_p N_p)^2}{6} \right)^{1/3}. \quad (9)$$

From this characteristic velocity, we then define the velocity space linking length,

$$\ell_v = \alpha_v \left(\frac{\Delta \bar{\rho} \pi G^3 (m_p N_p)^2}{6} \right)^{1/6}, \quad (10)$$

¹In reality, there is no such thing, but we need only an approximate value for the default linking length. For the same reason, we ignore the distinction between Δ and $\Delta + 1$.

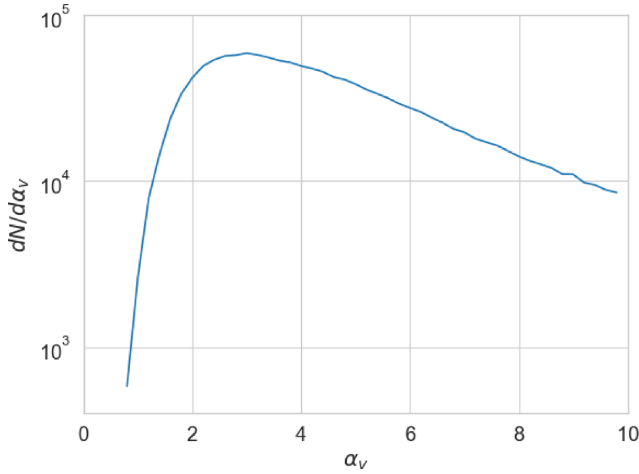


Figure 3. The distribution function of velocity linking length coefficients, α_v , that haloes exit the phase-space iteration with after achieving $E \leq 0$. Most haloes are bound with $\alpha_v = 10$ and so are omitted from the plot, as are those few that remain unbound for all $\alpha_v \geq 0.8$.

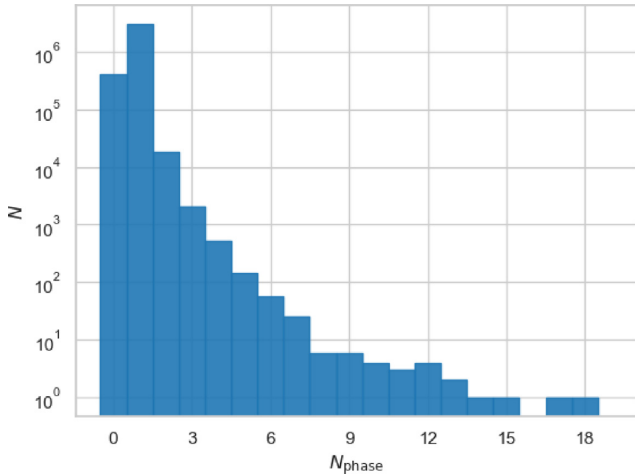


Figure 4. The number distribution of phase-space counterpart haloes for each spatially defined candidate halo. A phase-space number of 0 corresponds to a spurious spatial halo that has no phase-space counterpart.

where we have introduced the free parameter α_v whose value we vary as described below.

Using too small a value of α_v can lead to some of the smaller genuine haloes disappearing from the catalogue, while using too large a value fails to separate some obvious mergers. Additionally, as mentioned above, using a single linking length leads to a range of recovered overdensities for FOF haloes. For these reasons, we nominally set $\Delta = 200$ and loop over successively smaller values of α_v from 10 down to 0.8, exiting the loop and saving the properties of the halo(s) if at any stage $E < 0$. Should any haloes break into two or more pieces, then each is treated in the same manner. Should a halo reach $\alpha_v = 0.8$ with $E > 0$, it is included only in the final catalogue if it is required for another halo's persistence as described in Section 2.1.1.

The final α_v distribution is shown in Fig. 3 for those haloes that exit the iteration having reached $E \leq 0$, the number of haloes that result from each spatial counterpart is shown in Fig. 4, and the energy of the surviving haloes is shown in Fig. 5. As expected from Fig. 2, the vast majority (about 80 per cent) of haloes have

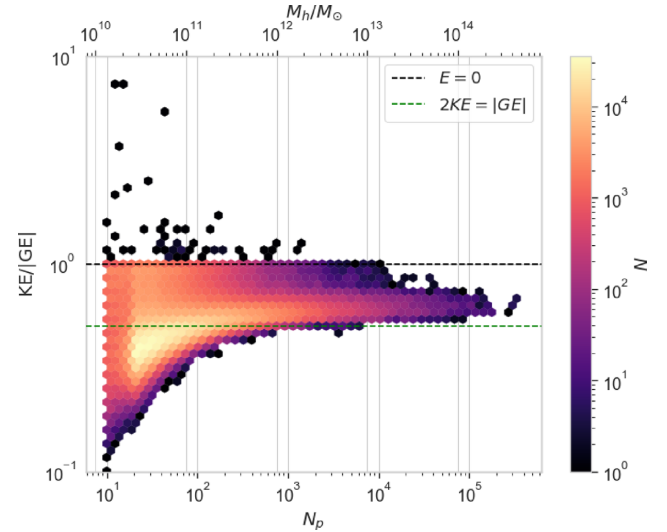


Figure 5. The energy distribution of all host haloes at all time-steps after applying phase-space splitting. As with Fig. 2, the colour of the markers signifies the density of points in that region of the distribution and $N_p < 20$ haloes are included only if they track the evolution of a halo, which at a previous time had $N_p \geq 20$.

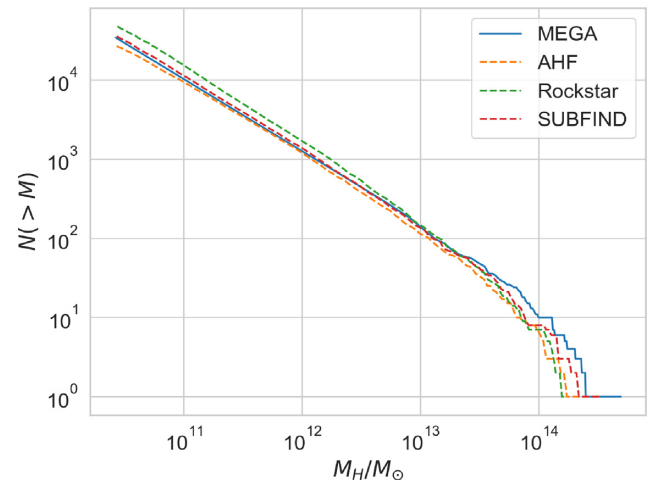


Figure 6. A comparison of the present day ($z = 0$) halo mass function for the halo catalogue produced from this simulation by the MEGA algorithm, Amiga Halo Finder (AHF), ROCKSTAR, and SUBFIND.

negative energy even when all particles are included and are thus unaffected by this move to phase space. However, 8 per cent only attain negative energy with a reduced velocity linking length, while 12 per cent of all candidate haloes are excluded because their particle count drops below 10 while their energy is still positive. Somewhat surprisingly, only 116 out of the original population of almost 1 million candidate haloes with $E > 0$ survive with 10 or more particles and positive energy at the end of this process – it is extremely efficient at removing rogue haloes. Of those 116, 59 are retained because they are the descendant of a real halo from the previous snapshot.

Fig. 6 shows the halo mass function at $z = 0$ in comparison to several other halo finders. Our method produces very similar halo numbers to SUBFIND and AHF at low masses, but our FOF method returns higher masses for the largest haloes. We stress, once again, that our haloes are not directly comparable to those in

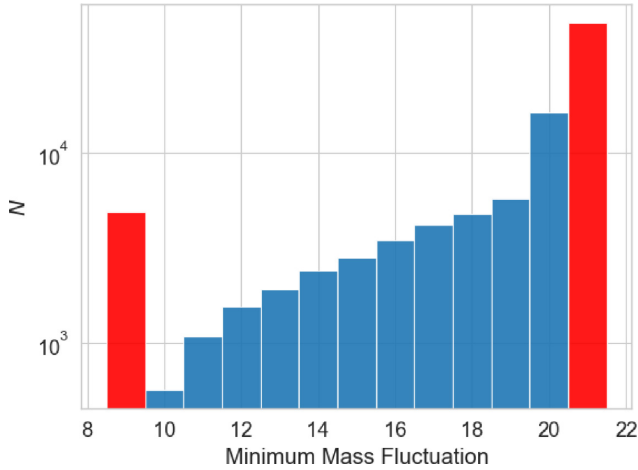


Figure 7. The distribution of minimum mass a halo reaches in its subsequent evolution once it has initially reached a particle mass of $N_p = 20$ or greater. Haloes included in the bin with mass less than <10 are those haloes that drop below the 10-particle limit and are no longer tracked. The haloes in the $N_p = 20$ bin are those that never drop below 20 particles once they reach that mass; those in the $N_p = 21$ bin are haloes that initially have a mass greater than 20 particles and never have $N_p \leq 20$ particles in their subsequent evolution.

SMT13 because we do not require individual particles to be bound. The justification for this looser definition comes in the improved behaviours of the resultant graphs and trees described in Section 3 below.

2.2 Construction of merger graphs

Given a halo catalogue, the construction of merger graphs in our method is elementary: we simply link together haloes in adjacent snapshots (output times) that have 10 or more particles in common. Recall that we require that all haloes, when they first appear, are at least 20 particles in size but may fluctuate down to 10 particles in their subsequent evolution. The distribution of minimum mass for a halo over its lifetime is shown in Fig. 7. About a quarter of haloes fluctuate below the 20-particle limit but only a small fraction (about 1 in 20) drops below 10 particles and so gives truncated branches that do not reach the present day.

2.3 Splitting merger graphs into merger trees

To facilitate comparison with previous work, we introduce a method to split graphs back into merger trees. A flow chart summarizing this procedure is shown in Fig. B2 of Appendix B.

Starting with the penultimate snapshot and working backwards in time, we consider the descendants of each halo. If there is more than one descendant, then we treat the halo as a candidate for splitting. The rationale is that the halo is a combination of two or more smaller haloes that are dynamically distinct but which have temporarily come within the FOF linking length of one another before subsequently splitting apart into multiple descendent haloes. We define the potential dynamically distinct split haloes by the particles that they have in common with each descendant and test their reality with the following constraints: that the split halo is the most massive candidate split halo, that it has negative energy, or that it occupies a distinct region in phase space relative to the most massive split halo. This overlap condition is given by the following

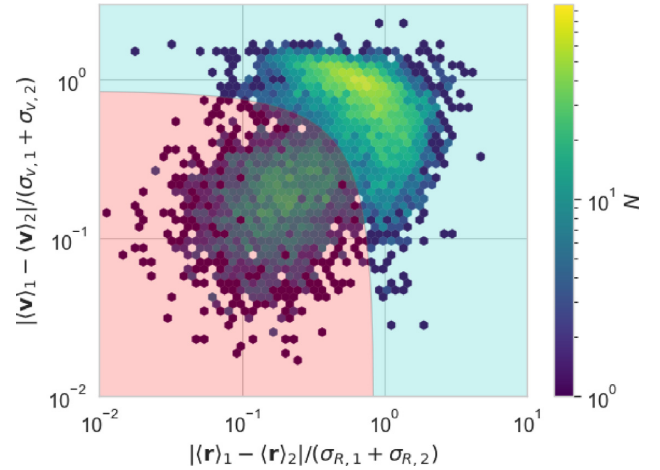


Figure 8. The separation of splitting candidate haloes (see Section 2.3) in real and velocity space where the candidate that is not the most massive split halo has $E > 0$. The boundary between coloured regions represents the condition we use (equation 11) to distinguish single haloes (red) from genuine overlaps (cyan).

expression in a similar approach to the method implemented by Han et al. (2018) to identify trapped subhaloes,

$$\frac{|(\mathbf{r})_1 - (\mathbf{r})_2|}{\sigma_{r,1} + \sigma_{r,2}} + \frac{|(\mathbf{v})_1 - (\mathbf{v})_2|}{\sigma_{v,1} + \sigma_{v,2}} \geq 0.85, \quad (11)$$

where $\langle \mathbf{r} \rangle$ and $\langle \mathbf{v} \rangle$ are the mean positions and velocities, σ_r and σ_v are the root-mean-square radii and velocities, and the subscripts 1 and 2 refer to the most massive split halo and the candidate split halo.

The choice of coefficient in equation (11) is motivated by Fig. 8, which shows the separations of split halo pairs in position and velocity space. Those at large separations in this diagram are genuine overlapping haloes that have failed to be separated by the phase-space halo identification method: these we accept as genuine distinct haloes. The long tail of low separation candidates, however, are most likely not distinct haloes but simply random associations of particles that come together to form the distinct descendant: for these, we omit the candidate split halo from our catalogue.

Once we have identified all genuine split haloes for any given halo, we allocate each a fraction of the original halo mass in proportion to the number of particles that it contains (for this purpose, the particles from rejected split halo candidates are associated with the most massive split halo, with which they overlap). Thus, the mass of the graph halo is equal to the sum of masses of all the tree haloes that it is split into.

In this way, after working back in time to the earliest haloes, the graph has become teased apart into separate trees, one for each halo in the graph at the final snapshot. The resultant number of haloes split from all candidate haloes is shown in Fig. 9. There are two histograms shown here: one for the number of bound split haloes that each halo is divided into, and another for the number of unbound split haloes, the larger numbers of unbound haloes are isolated to split haloes with a large total N_{split} . The main thing to take away is that even having a single unbound split halo is very rare (about 1 in 10^3) and having multiple ones even more so. To create a tree structure, some haloes need to be split into as many as 100 pieces, but these are almost all bound objects.

To further check that our algorithm does not simply tear apart real objects, we show the energy distribution of these split haloes in

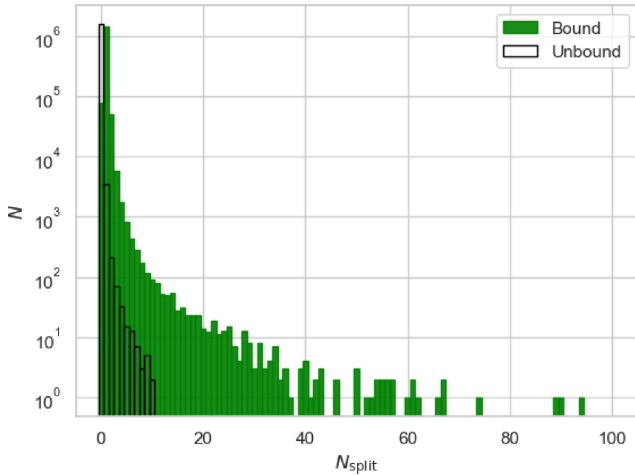


Figure 9. Histograms of the number of split haloes distinguishing between the number of instances of N_{split} split haloes that have negative and positive energy.

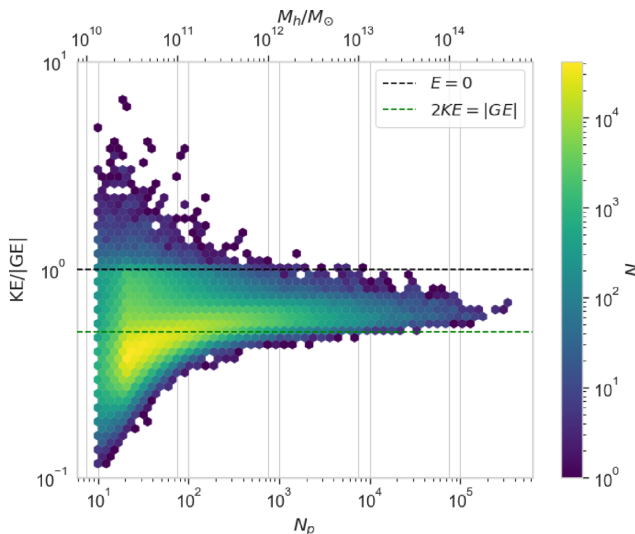


Figure 10. The energy distribution of all haloes at all time-steps after splitting graphs into their constituent trees, using only those particles in common between split haloes and their descendants.

Fig. 10. Note that some haloes with $N_p > 100$ that were previously bound (and even one with $N_p > 1000$) have now shown up as having $E > 0$ – that is because we are using in the energy calculation only those particles in common with the descendant. Of these split haloes with $E > 0$, the vast majority have a real ($E < 0$) progenitor. The remaining 18 exist for precisely two snapshots with $E > 0$ before linking to a real progenitor.

Overall, the splitting procedure seems to have done a reasonable job (and we will see in Section 3.1 below that the resultant tree has properties that match, or outperform, other methods).

2.4 Subhaloes: spatial nesting

The standard linking length of 0.2 times the mean inter-particle separation gives haloes that roughly correspond to the virialized region. The edges of such a region tend to be somewhat irregular and a certain amount of exchange of material with the surroundings is expected – hence, our use of merger graphs rather than merger

trees. In the current day Universe, at least, these tend to correspond to groups and clusters of galaxies: for that reason, we regard them as container vessels within which higher density galactic haloes will reside.

The overdensity of a halo (or that region of a halo) within which galaxies form is uncertain and probably redshift-dependent but will nevertheless be higher than the virial overdensity. Within this paper, we simply choose a linking length of 0.1 times the mean inter-particle separation to define such high-density subhaloes. Exactly as for the host haloes, we select candidate subhaloes in configuration space and then try successively smaller velocity linking lengths in phase space until the energy becomes negative, $E < 0$.

With this definition, all subhaloes are distinct from one another (no particles in common) and are spatially nested within the corresponding host halo.

3 RESULTS

We begin our results in Section 3.1 by looking at the properties of merger trees. That is not because we see merger trees as the way forwards but because we want to compare to previous work. We will show that our trees show similar mass fluctuations to other FOF-based methods in the SMT13 tests, but show fewer catastrophic failures and hence longer main branch lengths.

We then look in Section 3.2 at the properties of the graph as a whole. These show, as expected, smoother mass growth than for the trees.

Finally, in Section 3.3, we consider the properties of subhaloes, defined as higher density substructures within haloes.

3.1 Merger trees

Although the main purpose of this paper is to investigate the properties of merger graphs, we explained in Section 2.3 how to split the graphs back up into trees. This allows for comparison with previous work and, in particular, the study of SMT13: we reproduce the plots from that paper below.

We note two differences between our work and most of the SMT13 trees that should be borne in mind when making this comparison. First, we are defining haloes on the basis of FOF groups, rather than SO – our haloes are therefore irregular in shape and their mean density is rather poorly defined; to compare the results presented here to FOF-defined merger trees, see the results of Avila et al. (2014), particularly figs 3, 5, 7, and 8. Secondly, we require haloes to have a minimum of 20 particles when they are first created, but we allow them to fluctuate in mass down to 10 particles in their subsequent evolution – this has the effect of reducing the number of low-mass haloes that are brought into existence but then disappear with no descendants.

3.1.1 Main branch length

The first test of the quality of the trees that we consider is the main branch length: i.e. the number of simulation time-steps (snapshots) that a halo can be traced back into the past until it disappears. Fig. 11 shows that, grouped by halo mass (particle number) at the current day, and contrasted with the results from SMT13.

The most important thing to note is the absence of a tail extending to short main branch lengths in the upper panel and the reduced tail in the middle panel. That is to say, all massive haloes have existed for an extended period of time, as one might expect. That

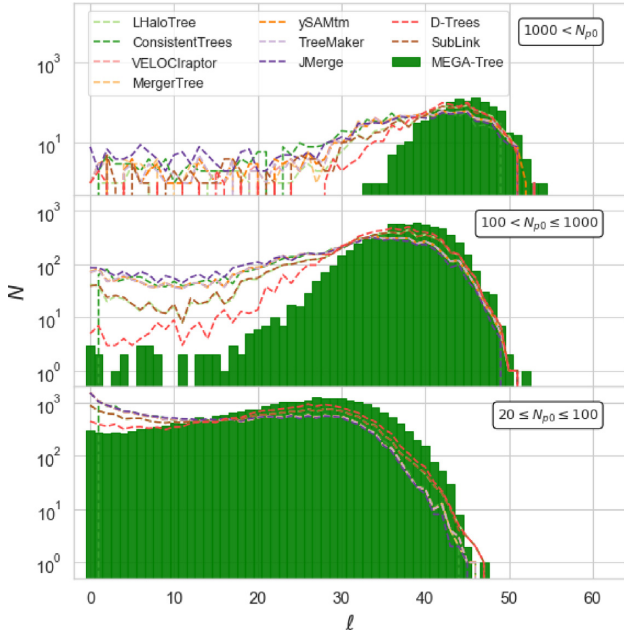


Figure 11. The main branch length of all trees rooted at the present day ($z = 0$). Here, ℓ is the number of snapshots that a tree’s main branch can be traced until it disappears. The upper, middle, and lower panels represent mass bins of $N_p > 1000$, $1000 \geq N_p > 100$, and $100 \geq N_p \geq 20$, respectively. The MEGA main branches are defined such that they are rooted at and end on haloes with $N_p \geq 20$; in between, they are allowed to fall below the 20-particle threshold but no lower than 10 particles.

is not true of the trees produced by the SMT13 study. In the middle histogram, for example, those show multiple haloes with more than 100 particles that come into existence only on the final snapshot. More worryingly, there are haloes with more than 1000 particles at the present day that exist for only one or two snapshots. Most likely, these are pathological cases whereby a large halo has split into two pieces, only one of which can be linked into the tree.

Even if we exclude these outliers, the median branch length is higher in our trees derived from the merger graphs: the maximum branch length is similar to that in SMT13 but the distribution is narrower. For the lower panel only, part of the improvement is due to our allowing haloes to fluctuate below 20 particles; we show the equivalent plots with a hard 20-particle lower limit in Appendix A.

3.1.2 Number of progenitors

Fig. 12 shows the number of direct progenitors for each halo in the trees. For MEGA-Tree, we restrict our analysis to haloes with $N_p \geq 20$ to facilitate comparison with the SMT13 results. It can be seen that MEGA-Tree is roughly consistent with those results but at the upper end of the progenitor number. The reason for this is that our looser definition of haloes, based on FOF rather than SO, leads to slightly more low-mass haloes (as can be seen in the lower panel of Fig. 11).

3.1.3 Mass growth

We next examine the mass growth along the main branch of the merger trees. Fig. 13 shows the slope of the logarithmic mass growth between any two main branch haloes with $N_p > 1000$ in adjacent snapshots. This statistic is defined to lie in the range of

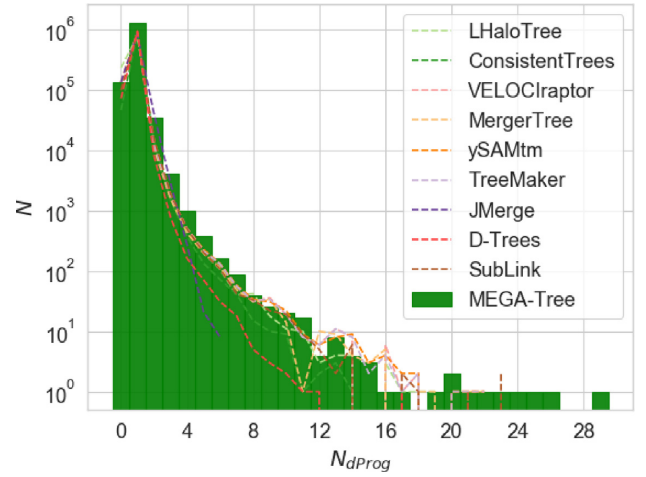


Figure 12. Histograms showing the number of direct progenitors of each halo at all time-steps. For MEGA-Tree, we include only haloes with $N_p \geq 20$, for comparison with previous work.

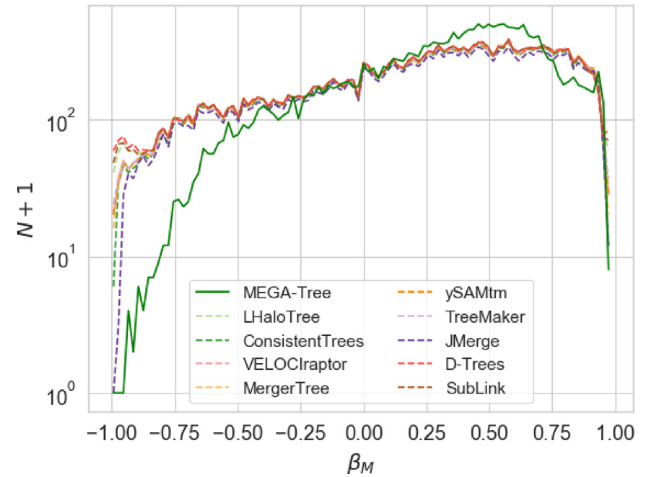


Figure 13. The distribution function of logarithmic mass growth along the main branches of all trees. This represents the slope of the mass evolution between all pairs of halo and descendant in a main branch, each with $N_p \geq 1000$.

$-1 \leq \beta_m \leq 1$ with

$$\beta_m(k-1, k) = \frac{\arctan(\alpha_m(k-1, k))}{\pi/2}, \quad (12)$$

where the logarithmic mass growth is defined as

$$\frac{d \log(M)}{d \log(t)} \approx \alpha_m(k-1, k) = \frac{(t_k + t_{k-1})(M_k - M_{k-1})}{(t_k - t_{k-1})(M_k + M_{k-1})}. \quad (13)$$

Thus, $\beta_m \sim -1$ represents a catastrophic decrease in mass, and $\beta_m \sim +1$ an abrupt increase in mass.

This statistic shows that it is much more common for haloes in MEGA-Tree to experience modest mass growth between snapshots than those from SMT13 trees, and that the fraction of haloes that decrease in mass is lower. These differences between SMT13 algorithms and MEGA-Tree match those found in Avila et al. (2014), who showed that they can almost entirely be attributed to the difference in halo definition: FOF here compared to SO in SMT13. Another difference, which is not simply due to the use of

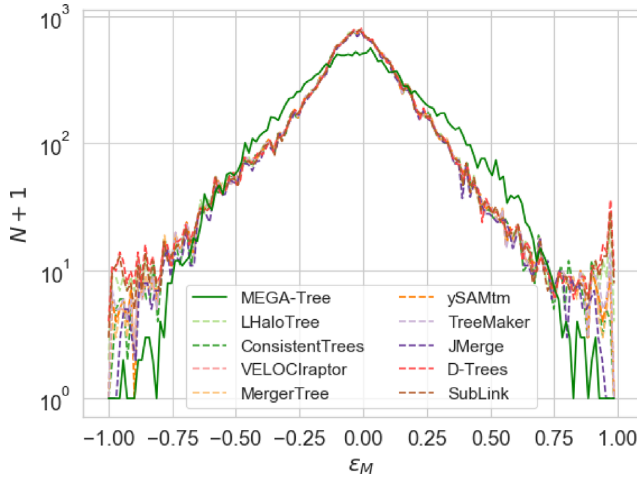


Figure 14. The mass fluctuation of all sets of three consecutive temporally linked haloes in the main branch of a tree, each with $N_p \geq 1000$.

FOF, is that the number of catastrophic changes in mass $|\beta_m| \geq 0.75$ is reduced relative to SMT13 algorithms.

Another method to categorize the mass growth is the fluctuation in the rate of mass growth: this is presented in Fig. 14 for all successive main branch haloes with $N_p > 1000$. The mass fluctuation is given by

$$\epsilon_m = \frac{\beta_m(k, k+1) - \beta_m(k-1, k)}{2}. \quad (14)$$

At first sight, the difference between Figs 13 and 14 seems odd: the former shows a greater tendency for positive mass growth in MEGA-Tree than in SMT13 but a broader spread of fluctuations in the rate of growth. However, we have checked that this behaviour is real, and a similar effect was again observed in Avila et al. (2014) for FOF-defined haloes as compared to the SO-defined haloes in SMT13.

So in summary, our MEGA-Tree trees show similar fluctuations in mass to previous algorithms but far fewer catastrophic failures, i.e. sudden appearance or disappearance of massive haloes.

3.2 Merger graphs

We now investigate the main focus of this work, merger graphs. Where possible, we compare statistics with those produced by trees, both in this work and in SMT13.

3.2.1 Main branch length

Once again, the first quality of graphs we measure is the main branch length, grouped by halo mass (particle number) at the current day and contrasted with the results from SMT13 in Fig. 15.

As with the trees, the median branch length is higher than that in SMT13 with a narrower distribution and a similar maximum branch length. This can be attributed to our merger graphs being better behaved at low particle numbers, which is again evident in the lower panel of Fig. 15 with no sharp upturn at short branch lengths relative to the upturn on SMT13 trees at the same point.

More importantly, catastrophic failures in the evolution of haloes with $N_p > 100$ are completely eliminated. Unlike for trees, the graph does not require decisions to be made about which descen-

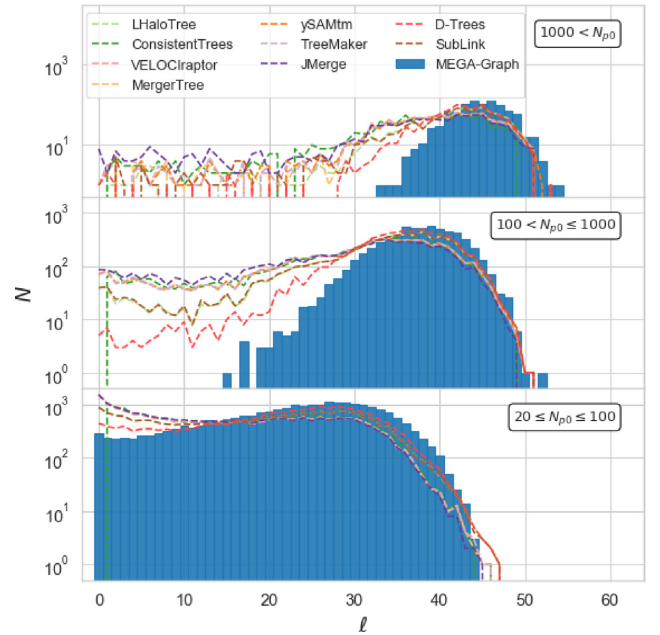


Figure 15. The main branch length of all graphs rooted at the present day ($z = 0$). As in Fig. 11, ℓ is the number of steps or generations possible ‘walking’ down the tree, with the lower, middle, and upper panels representing mass bins of $20 \leq N_p \leq 100$, $100 < N_p \leq 1000$, and $1000 \leq N_p$, respectively. Here, the MEGA main branches are defined such that they are rooted at generations of the graph with at least one halo with $M \geq 20$ and end on haloes with $M \geq 20$; in-between haloes are allowed to fall below the 20-particle threshold but no lower than 10 particles.

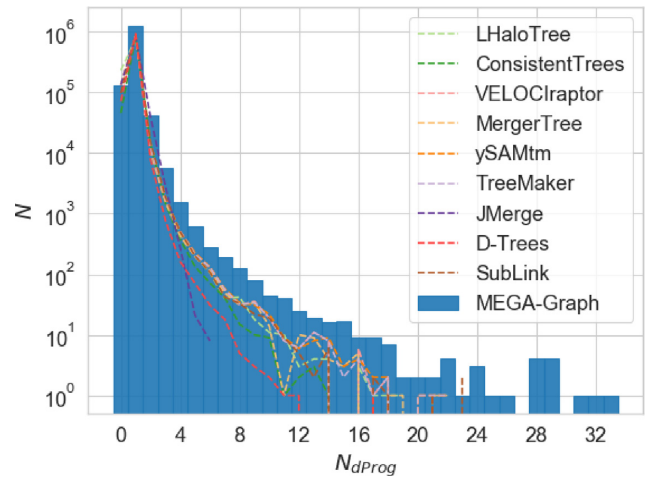


Figure 16. Histograms showing the number of direct progenitors of each halo in all time-steps. Once again, for the SMT13 algorithms, this includes all haloes within their trees; for MEGA-Graph, this includes all haloes with $N_p \geq 20$.

dants to keep and so there is no possibility that halo will become detached from the graph.

3.2.2 Number of progenitors and descendants

The number of direct progenitors of MEGA-Graph haloes is shown in Fig. 16. Unsurprisingly, this is similar to the result for haloes in the MEGA-Tree, shown in Fig. 12.

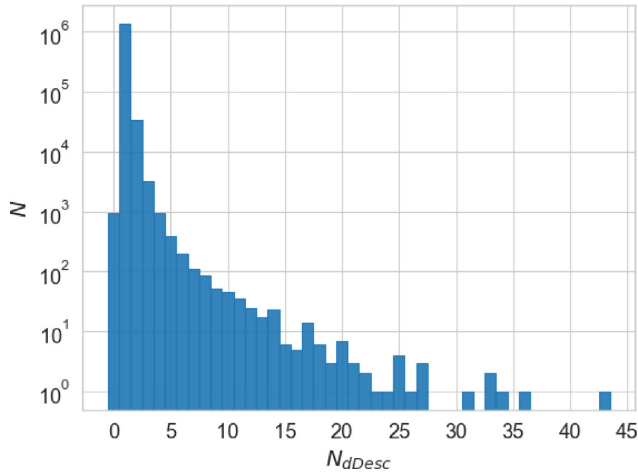


Figure 17. Histogram showing the number of direct descendants with $N_p \geq 10$ for all MEGA-Graph haloes in all snapshots.

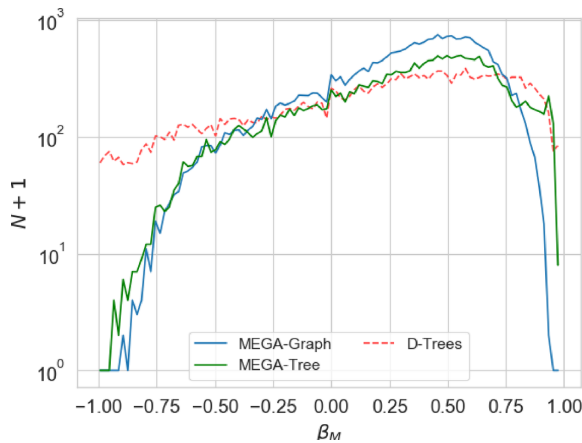


Figure 18. The distribution of logarithmic mass growths between graph generations compared to the main branch of MEGA-Tree and those of the SMT13 algorithm D-Trees. This plot represents the slope of the mass evolution between all pairs of generations with a total number of particles $N_p \geq 1000$. For the trees, this is all pairs of halo and descendent in a main branch both with $N_p \geq 1000$.

More interesting is the number of direct descendants, shown in Fig. 17. For a tree, this is mostly 1, except for the rare occasions when a halo goes missing, in which case it is 0. Notice that the MEGA-Graph algorithm has not eliminated 0 descendants (end haloes) entirely: these are those haloes, as discussed in Section 2.2 and shown in Fig. 7, whose mass eventually dwindles below a particle number of 10.

3.2.3 Smoothness of mass growth

It is when we look at the smoothness of the mass growth that the value of graphs over trees becomes clear. Fig. 18 shows that the mass growth is much more strongly biased towards positive values even than the MEGA-Trees and that the abrupt increases and decreases of mass have been entirely eliminated. Similarly, in Fig. 19, we see that mass fluctuations are smoother in the MEGA-Graphs than in any tree. While not entirely surprising, this does reinforce the idea

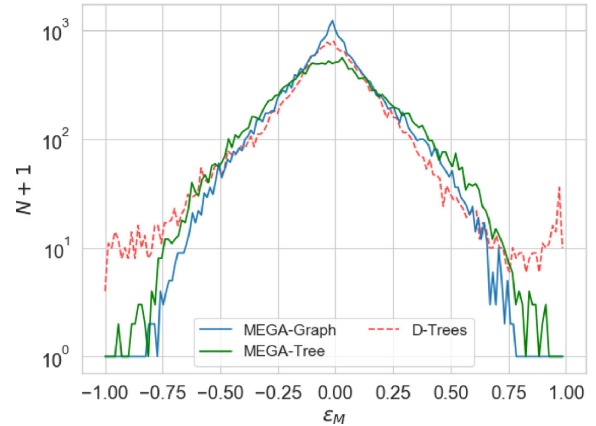


Figure 19. The mass fluctuation of all sets of three consecutive graph generations with $N_p \geq 1000$. Again, we present MEGA-Tree and D-Trees for comparison where three consecutive main branch haloes have $N_p \geq 1000$.

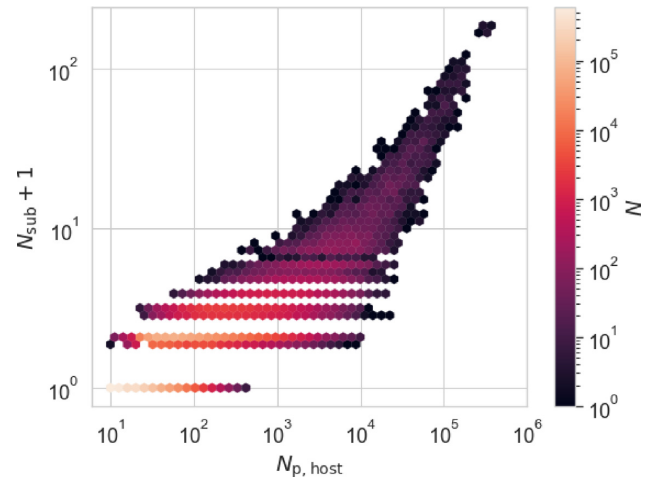


Figure 20. The occupancy as a function of host halo mass in MEGA for all subhaloes at all snapshots. The cluster of points at the tip of the distribution is located in a single halo at multiple sequential time-steps.

that graphs provide a more complete framework for following the growth of structure than do trees.

3.3 Substructure

In this section, we look at the properties of subhaloes within our host haloes. As a reminder, for the purposes of this paper, subhaloes are simply defined to be haloes defined in exactly the same way as for the host haloes, but with linking lengths in both position and velocity corresponding to eight times the overdensity.

We define the central subhalo to be the one whose centre of mass is closest to the centre of mass of the enclosing host halo – in nine out of 10 cases, this is the most massive subhalo. Other subhaloes are referred to as satellite subhaloes.

3.3.1 Halo occupation distribution

Fig. 20 shows the relationship between the number of subhaloes, N_{sub} , and the particle number in the host halo. By far, the most common values of N_{sub} are 0 and 1: the former correspond to small

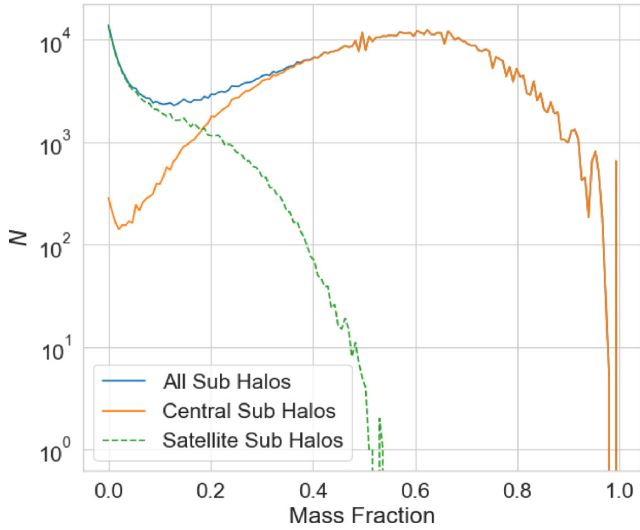


Figure 21. Histograms of subhalo mass fractions, the fraction of a host halo’s mass held within the central subhalo and all satellite subhaloes. The spikiness of the curves, most evident for the central subhaloes, comes from the discreteness of the particle number in low-mass haloes.

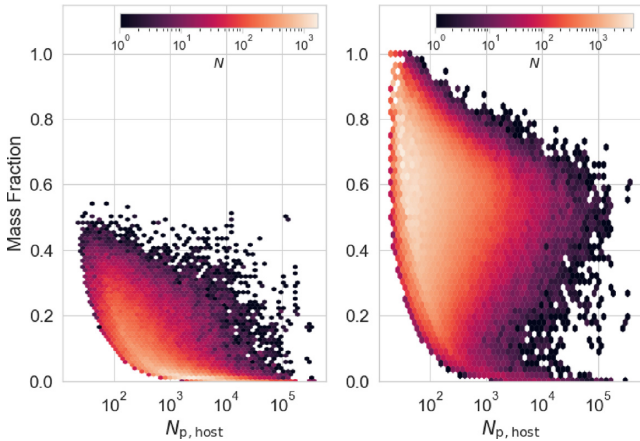


Figure 22. The mass fraction from Fig. 21 plotted as a function of host halo mass. The left-hand panel and right-hand panel show the mass fraction of satellite subhaloes and central subhaloes, respectively.

haloes for which substructure cannot be resolved and the latter to haloes that have a single density peak. However, we also see occupancy numbers of decreasing frequency, as high as $N_{\text{sub}} = 140$.² Of course, the number of subhaloes is primarily limited by the resolution of the simulation, i.e. the number of particles in each halo.

3.3.2 Subhalo mass fractions

Figs 21 and 22 show histograms of the contribution to the mass of a halo from substructures, and a breakdown by halo mass, respectively. Note that these plots show only those haloes that possess subhaloes: this is about half of the total. The requirement that a subhalo has a minimum mass of 20 particles leads to the

²Note that the clump of points with $N_{\text{sub}} > 100$ in Fig. 20 all correspond to haloes at different snapshots along the main branch of a single graph.

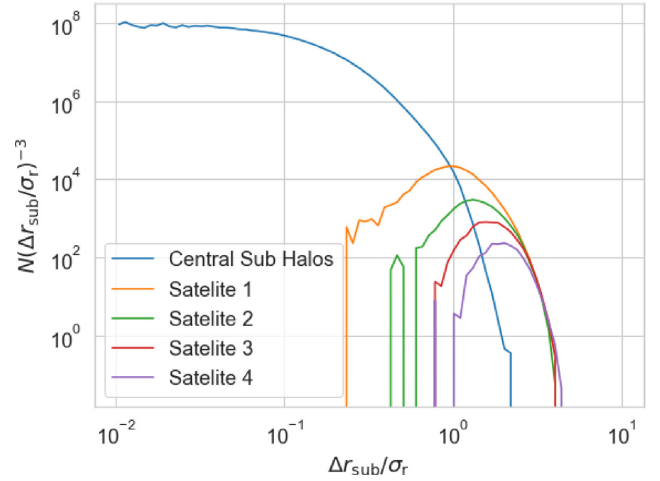


Figure 23. Space density of the centrality of subhaloes for all host haloes and subhaloes with $N_p \geq 10$ in all snapshots. The centrality is the separation of a subhalo’s centre of mass from the centre of mass of its enclosing host halo, Δr_{sub} , divided by the root-mean-square size of the host halo, σ_r . Plotted here are the central subhalo and then the next 4 subsequent satellite haloes in increasing radius from the centre of the host halo.

lower locus seen in the plots and biases the mass fractions upwards in the lowest mass haloes. Overall, the mass fraction of the central subhalo is about 50 per cent but with a large dispersion. That leaves little room for satellite subhaloes that mostly have very small mass fractions.

The right-hand panel in Fig. 22 shows a branch of very low-mass fractions for the central halo – that is because there are a small number of merging systems for which the most massive subhalo is not located at the centre of the FOF group. Were we to choose to define the ‘main’ subhalo as the most massive one, then this low-mass fraction branch would disappear, but we prefer not to do that for the reason discussed in the next subsection.

3.3.3 Subhalo centrality

One of the advantages of defining all subhaloes, including the central subhalo as being distinct from the enclosing host halo, is that we can determine whether or not the two have common dynamics, or whether the central subhalo is displaced or is moving with respect to the host: this is shown in Figs 23 and 24, respectively.³

We can see that central haloes completely dominate at small radii and remain the most populous type of subhalo all the way out to σ_r , where σ_r is the root-mean-square size of the host halo. Most likely, satellite subhaloes are disrupted at smaller radii, although there may be some that survive but go undetected by our halo-identification algorithm. The median offset of the central subhalo from the centroid of the host halo is about $0.27 \sigma_r$.

In velocity space, there is no such restriction and the density of satellite subhaloes remains constant, albeit much lower than that of the main subhalo, all the way in to relative velocities of zero. The median speed of the central subhalo relative to that of the host halo is about $0.12 \sigma_v$.

³The scaling of number count with inverse volume in either configuration or velocity space in Figs 23 and 24 gives the density of haloes.

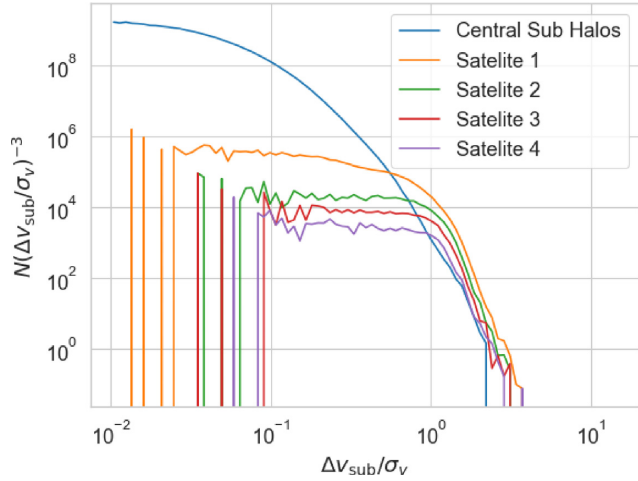


Figure 24. Velocity-space density of the relative speed of central and satellite subhaloes for all host haloes and subhaloes with $N_p \geq 10$ in all snapshots. The relative speed is the magnitude of the difference in velocity of a subhalo from that of the host halo, Δv_{sub} , divided by the velocity dispersion of the host halo, σ_v . Plotted here are the central subhalo and then the next four subsequent satellite haloes in increasing radius from the centre of the host halo.

4 CONCLUSIONS AND DISCUSSION

In this paper, we advocate the use of merger graphs rather than merger trees in order to follow the development of structure in the Universe. Our method leads to two significant advantages compared to previous approaches:

- (i) The elimination of catastrophic failures in merger histories whereby massive haloes either disappear or are created without progenitors.
- (ii) The reduction of mass fluctuations during the growth of structure.

We identify haloes by a two-step process: first using FOF to locate over-dense regions in real (configuration) space and then by running FOF a second time, in phase space, with a decreasing velocity-space linking length, until a candidate halo either has negative total energy or drops below the 10-particle limit. Although this method does an excellent job at distinguishing interacting but unbound haloes that will later separate, it is not perfect. Therefore, in addition, a small fraction of haloes with positive energy are retained if they are descendants of (have 10 or more particles in common with) real haloes in the previous snapshot.

To aid comparison with previous work (Srisawat et al. 2013, SMT13), we also generate merger trees by splitting haloes with multiple descendants into two or more pieces. We do not require that the pieces have negative energy, but we do require that they occupy distinct regions of phase space as an indicator that they are genuine structures. Our main results are:

- (i) The distribution of main branch length (length of time that a halo exists for) is much more peaked than in other methods from SMT13 and, in particular, shows many fewer short lengths that are indicative of catastrophic failures (haloes being created or disappearing out of nothing).
- (ii) The number of progenitors is similar to the SMT13 algorithms.
- (iii) The mass growth along the main branch is slightly more peaked and biased to positive values, but the distribution of mass

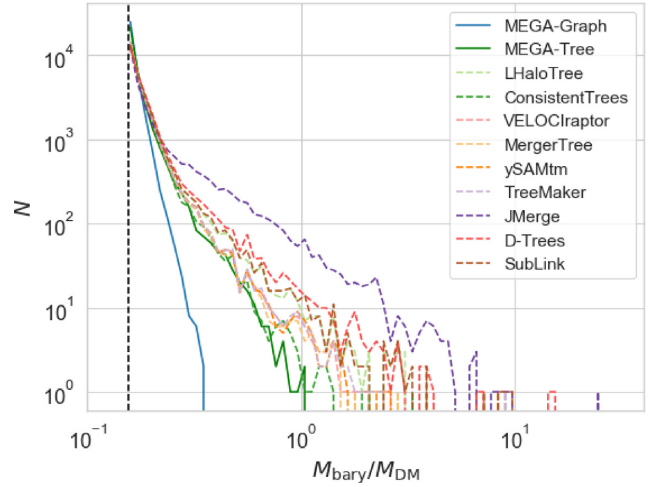


Figure 25. Histograms showing the baryon fraction within current day haloes for merger trees and merger graphs. The vertical, dashed line shows the mean baryon fraction of the Universe.

fluctuations is slightly broader: this is similar to the result found by Avila et al. (2014), who showed that the difference arises from the difference between FOF haloes and the use of SO as in SMT13.

The main purpose of this paper is to look at the properties of merger graphs, with the following conclusions:

- (i) As for our derived merger trees, the advantages of increased branch lengths compared to previous methods from SMT13 are retained.
- (ii) In addition, the graphs show smoother mass growth than any tree algorithm.
- (iii) The most common number of descendants for a halo is 1, as one might expect, but can rise as high as 43.

When modelling star formation in galactic haloes, it is often assumed that the baryon content of haloes can never decrease: this is to preserve metallicity (see e.g. Henriques et al. 2020). That means that the baryon content of a halo is determined by the ratio of the maximum mass along its past history to the current mass. Fig. 25 shows histograms for the baryon fraction in current day haloes derived from two different trees and MEGA-Graph, where the baryon mass is set equal to the mean baryon density parameter of the Universe (set to 0.0155 here) times this maximum mass. In each case, we have strong peaks at the desired universal value but with a tail of mass ratios extending towards the right: for the best of the SMT13 algorithms, this can even give baryon masses that exceed the total mass of the halo! The MEGA-Tree algorithm performs better than any of those from SMT13, but clearly MEGA-Graph greatly improves on any of the trees.

The haloes that we identify with a linking length of 0.2 times the mean separation we call host haloes: we then generate high-density ‘subhaloes’ through the same procedure as for host haloes but using half the spatial linking length. This guarantees nesting of subhaloes within main haloes. The lowest mass host haloes have either a single subhalo or none at all, with the number of subhaloes increasing with host halo mass up to, for the SMT13 simulation, over 100.

We divide the subhaloes up into central subhaloes (those closest to the centre of each host halo) and satellite subhaloes. Typically, a fraction of about 0.5–0.6 of the mass is contained in the central subhalo. As might be expected, the central subhalo is generally located towards the centre of the enclosing main halo, with a

median separation between their centres of about $0.27 \sigma_r$, where σ_r is the root-mean-square size of the host halo. Similarly, most central subhaloes are moving at roughly the same velocity as their hosts. In a small fraction of cases, however, corresponding to merging systems, there are substantial offsets between the positions and/or velocities of central subhaloes relative to their hosts.

In the context of galaxy formation, then we associate haloes with the potential wells (extended galactic haloes, groups, and clusters) within which baryons will accumulate. The galaxies themselves will form within the higher density subhaloes. The central subhalo is usually, but not always, located at the centre of the host halo and will be the natural site of accretion for cooling coronal gas: where there is a large offset, then that is an indicator of a merger and accretion could be suspended.

Thus, we anticipate that our merger graphs will provide a better backbone for galaxy formation models than existing merger trees in three respects:

- (i) They eliminate critical failures in the halo growth history: massive haloes with either no progenitors or no descendants.
- (ii) They provide smoother mass growth for the graph as compared to individual trees.
- (iii) They distinguish between the central subhalo and the host halo (unlike other methods) and thus provide a better treatment of mergers.

For existing semi-analytic models that rely on merger trees rather than merger graphs, then the MEGA-Tree algorithm greatly improves with respect to other trees on the first of these properties and also satisfies the third. (We note, however, that, unlike the graphs, trees cannot be constructed on-the-fly as MEGA-Tree requires that we start with the final snapshot and work back in time.)

We intend to investigate the use of MEGA-Graph and MEGA-Tree for galaxy formation modelling in a future paper.

ACKNOWLEDGEMENTS

The authors contributed in the following way to this paper: P. A. T. originated and supervised the project; W. J. R. did the vast majority of the coding and produced the figures; P. A. T. and W. J. R. together drafted the paper; C. S. worked on an early version of the project, chipped in with useful suggestions from time to time, and helped to improve the paper.

W. J. R. was jointly supported by a Royal Astronomical Society summer student bursary and by a Junior Research Scholarship from the University of Sussex for a substantial part of this project. P. A. T. acknowledges support from the Science and Technology Facilities Council (grant number ST/P000525/1). C. S. acknowledges support by a Grant of Excellence from the Icelandic Research Fund (grant number 173929-051).

This project made extensive use of data from the Sussing Merger Trees project (Srisawat et al. 2013), which is available upon request from the authors: we would like to thank everyone who contributed to that project.

We have benefitted greatly from the publicly available programming language `python`, including the `numpy`, `matplotlib`, `scipy`, and `h5py` packages. With thanks to Thompson (2014),⁴ used to read GADGET binary files.

This work used the DiRAC@Durham facility managed by the Institute for Computational Cosmology on behalf of the STFC DiRAC

HPC Facility (www.dirac.ac.uk). The equipment was funded by BEIS capital funding via STFC capital grants ST/K00042X/1, ST/P002293/1, ST/R002371/1, and ST/S002502/1, Durham University, and STFC operations grant ST/R000832/1. DiRAC is part of the National e-Infrastructure.

In addition, much of the early analysis was undertaken on the APOLLO cluster at Sussex University.

REFERENCES

- Avila S. et al., 2014, *MNRAS*, 441, 3488
 Behroozi P. S., Wechsler R. H., Wu H.-Y., 2013a, *ApJ*, 762, 109
 Behroozi P. S., Wechsler R. H., Wu H.-Y., Busha M. T., Klypin A. A., Primack J. R., 2013b, *ApJ*, 763, 18
 Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, *ApJ*, 292, 371
 Elahi P. J., Cañas R., Poulton R. J. J., Tobar R. J., Willis J. S., Lagos C. d. P., Power C., Robotham A. S. G., 2019a, *PASA*, 36, e021
 Elahi P. J., Poulton R. J. J., Tobar R. J., Cañas R., Lagos C. D. P., Power C., Robotham A. S. G., 2019b, *PASA*, 36, e028
 Gill S. P. D., Knebe A., Gibson B. K., 2004, *MNRAS*, 351, 399
 Han J., Jing Y. P., Wang H., Wang W., 2012, *MNRAS*, 427, 2437
 Han J., Cole S., Frenk C. S., Benitez-Llambay A., Helly J., 2018, *MNRAS*, 474, 604
 Henriques B. M. B., Yates R. M., Fu J., Guo Q., Kauffmann G., Srisawat C., Thomas P. A., White S. D. M., 2020, *MNRAS*, 491, 5795
 Jiang L., Helly J., Cole S., Frenk C., 2014, *MNRAS*, 440, 2115
 Jung I., Lee J., Yi S. K., 2014, *ApJ*, 794, 74
 Knebe A. et al., 2011, *MNRAS*, 415, 2293
 Knollmann S. R., Knebe A., 2009, *ApJS*, 182, 608
 Komatsu E. et al., 2011, *ApJS*, 192, 18
 Lacey C., Cole S., 1993, *MNRAS*, 262, 627
 More S., Kravtsov A. V., Dalal N., Gottlöber S., 2011, *ApJS*, 195, 4
 Press W. H., Schechter P., 1974, *ApJ*, 187, 425
 Rodriguez-Gomez V. et al., 2015, *MNRAS*, 449, 49
 Springel V. et al., 2005, *Nature*, 435, 629
 Springel V., 2005, *MNRAS*, 364, 1105
 Springel V., White S. D. M., Tormen G., Kauffmann G., 2001, *MNRAS*, 328, 726
 Srisawat C. et al., 2013, *MNRAS*, 436, 150
 Thomas P. A. et al., 2015, preprint ([arXiv:1508.05388](https://arxiv.org/abs/1508.05388)) (TOT15)
 Thompson R., 2014, pyGadgetReader: GADGET snapshot reader for python, (ascl:1411.001)
 Tweed D., Devriendt J., Blaizot J., Colombi S., Slyz A., 2009, *A&A*, 506, 647

APPENDIX A: OMISSION OF 10-PARTICLE HALOES

We wish to emphasize that the strength of both MEGA-Tree and MEGA-Graph is not due to the mass limit of $N_p > 10$. Instead, their strength comes from the encoding of information from an arbitrary number descendants, inherent in the construction of a graph, and subsequent teasing apart into trees. To this end, we have regenerated our results using only haloes with $N_p \geq 20$ for more direct comparison with the results from SMT13. We present the main branch lengths in Figs A1 and A2 for the trees and graphs, respectively.

Strikingly, in both the graphs and trees, we retain the behaviour observed in Figs 11 and 15 in the middle and upper panels. In the case of Fig. A2, this definitively shows the robustness of the merger graph approach we advocate. However, for MEGA-Tree, the retention of the improved behaviour in these mass bins may come as a surprise. Once again, this improvement can be attributed to the inclusion of descendent information in merger graphs, which

⁴<https://bitbucket.org/rthompson/pygadgetreader/>

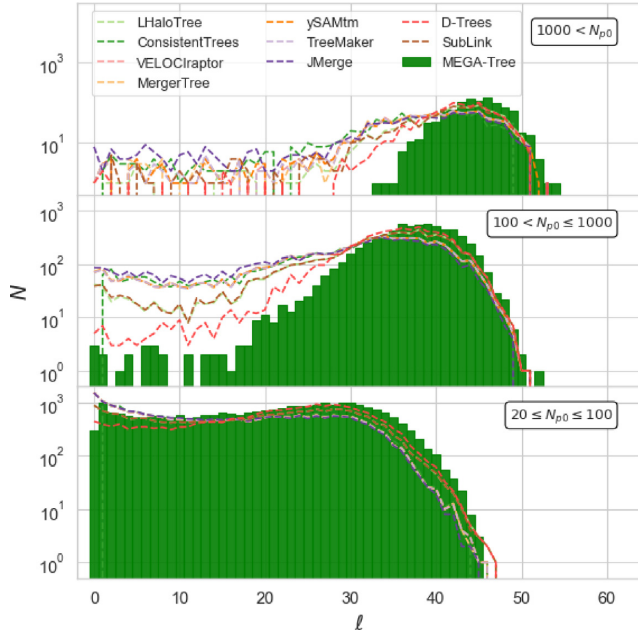


Figure A1. The main branch length of all trees rooted at the present day ($z = 0$) omitting $N_p < 20$ haloes. ℓ is the number of snapshots that a tree's main branch can be traced until it disappears. The upper, middle, and lower panels represent mass bins of $N_p > 1000$, $1000 \geq N_p > 100$, and $100 \geq N_p \geq 20$, respectively.

is then inherited by the trees via the splitting of haloes with multiple descendants.

In the lower mass bin, however, the graph approach does not provide a significant improvement but gives results similar to the [SMT13](#) algorithms, with the exception of D-Trees, which tracks fluctuations at the low-mass limit better than others, allowing snapshots to be skipped when looking for progenitors. Omitting these $N_p < 20$ haloes removes our mechanism to compensate for downwards mass fluctuations in low-mass haloes.

To have well-behaved main branch lengths in the low-mass regime, it is therefore imperative to have some mechanism to track fluctuations around the mass limit of a halo definition, be it snapshot

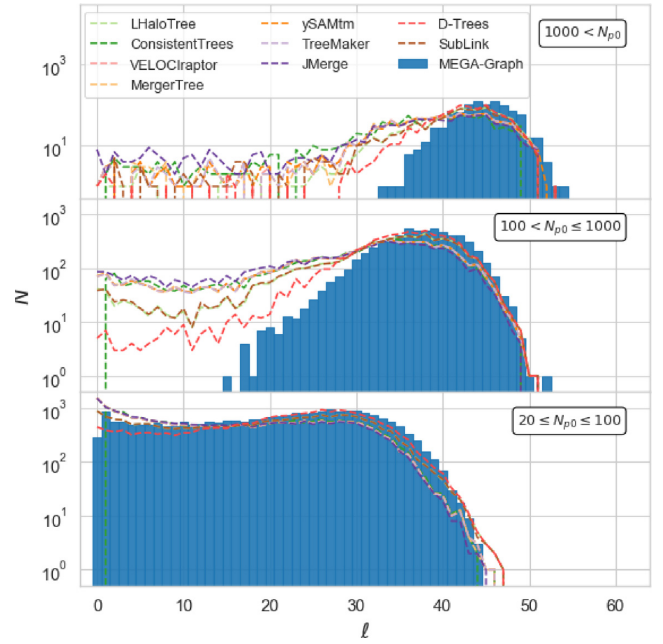


Figure A2. The main branch length of all graphs rooted at the present day ($z = 0$) omitting $N_p < 20$ haloes. ℓ is the length of a main branch, and the lower, middle, and upper panels represent mass bins of $20 \leq N_p \leq 100$, $100 < N_p \leq 1000$, and $1000 \leq N_p$, respectively. The MEGA-Graph main branches are defined such that they are rooted at generations of the graph where the most massive halo has a mass bounded by the upper and lower limits of the respective mass bin.

skipping as utilized in D-Trees or tracking haloes below the halo mass definition as employed here.

APPENDIX B: FLOW CHARTS OF THE ALGORITHM

Here, we present flow charts detailing the more complex elements of MEGA. We have omitted the method for merger graph construction as this is very straightforward, as described in Section 2.2. The halo finder and tree-splitting algorithms are presented in Figs [B1](#) and [B2](#), respectively.

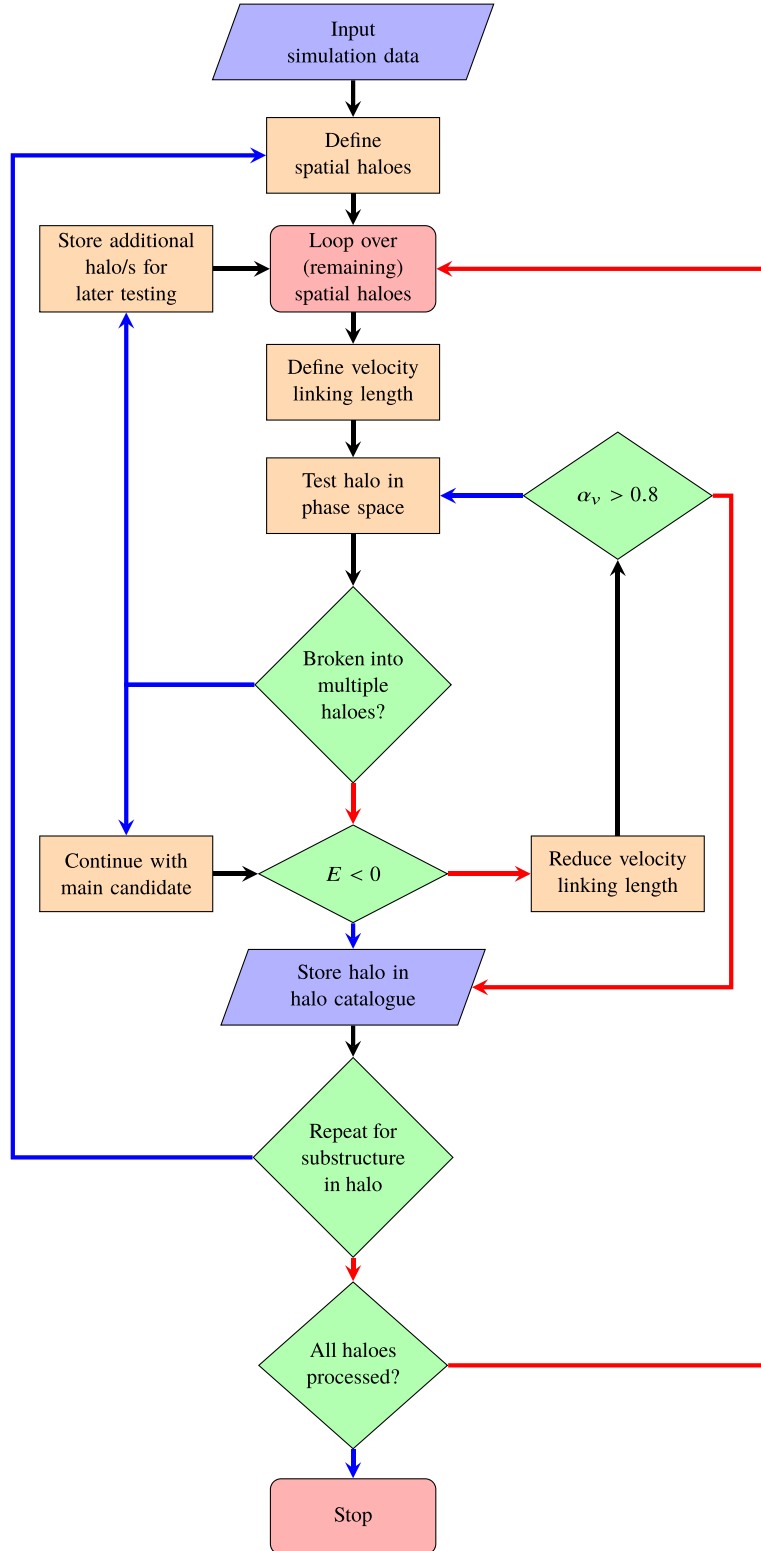


Figure B1. A flow chart of the halo finder algorithm. This process is performed on each snapshot independently, producing the halo catalogue, which is later pruned at the merger graph construction phase: eliminating haloes with $E > 0$ and $N < 20$ that do not have any progenitors. In the flow chart, blue trapeziums represent I/O, red rounded boxes represent loops, orange boxes represent processes, and green diamonds represent logic. Blue arrows represent True/Yes, while red arrows represent False/No.

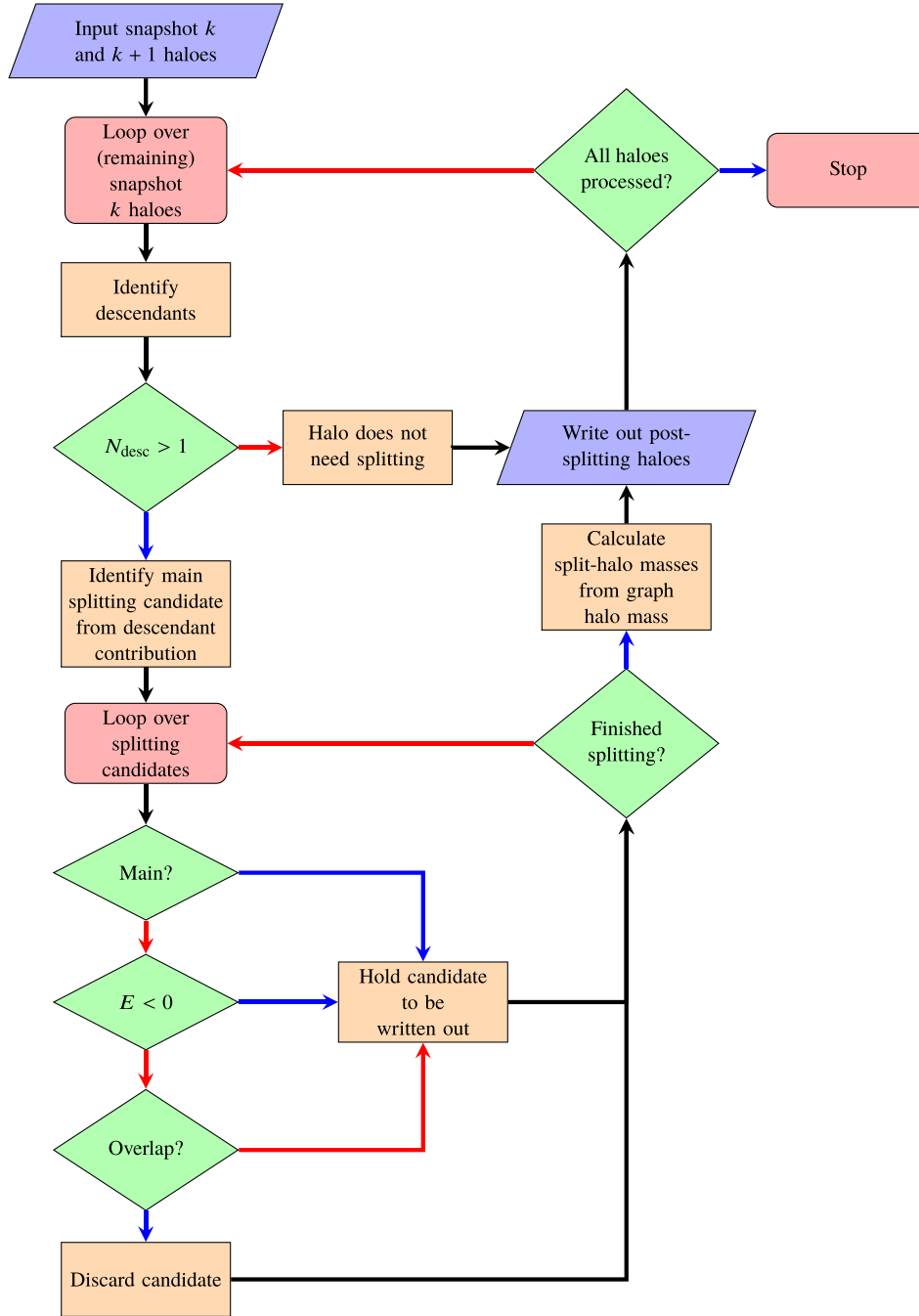


Figure B2. A flow chart of the graph-splitting algorithm used to build merger trees from merger graphs. This is done on a snapshot-by-snapshot basis, starting at the penultimate snapshot and working back in time. In the flow chart, blue trapeziums represent I/O, red rounded boxes represent loops, orange boxes represent processes, and green diamonds represent logic. Blue arrows represent True/Yes, while red arrows represent False/No.

This paper has been typeset from a \LaTeX file prepared by the author.